Vision System for Disabled People Using Pattern Matching Algorithm

Armen Petrosyan
Institute of Informatics and Automation Problem, National Academy of Sciences of Armenia
Yerevan, Armenia
e-mail: armenpet@ipia.sci.am

ABSTRACT

In last years there are some actual tasks to detect and recognize the human faces. It could be used either for recognizing the person or just for motion detection and finding the position of face in the image.

In this paper is described one solution regarding face position detection and finding of certain parts of face in images. The solution works on images acquired from several types of cameras, such as GigE Cameras, USB/Web Cameras, Smart Cameras (Real-Time Targets).

The system consists of two parts – acquisition part and image processing part.

Keywords

Pattren Recognition and Image Processing, Computer science, informatics, LabVIEW.

1. INTRODUCTION

There is an opportunity and idea to create a system to support and help disabled people, who can't use mouse of the computer and need additional accessibilities to control the mouse cursor. So, one way is to use cameras for it. Via cameras it will be possible to control human face motion and the changes of face parts (eye, mouth), then move and click mouse cursor - connected by certain changes. The frames (images), which come from camera acquisition, will pass some image processing, then using pattern matching algorithm will be implemented detection of face motion and then start move mouse cursor in the direction from initial point or do clicking of mouse. Actually it is control of on-screen mouse cursor by tracking the face of a person.

In this paper we will introduce pattern matching algorithm and the tools, which used to implement the task in real life [2].

Pattern Matching algorithms are some of most important functions in machine vision, because of their use in varying applications as alignment, gauging, inspection [1]. Pattern matching is an algorithm for locating regions of a grayscale image that match a known reference pattern, also referred to as a model or template. The template is an idealized representation of a feature in the image.

When using pattern matching, you create a template that represents the object for which you are searching. Your machine vision application then searches for instances of the template in each acquired image, calculating a score for each match. This score relates how closely the template resembles the located matches.

Pattern matching finds template matches regardless of lighting variation, blur, noise, and geometric transformations such as shifting, rotation, or scaling of the template.

Pattern matching is a method of identifying features in an image that match a smaller template image (that is, the "pattern" to be matched). The process involves two phases: an off-line learning phase in which the template is processed, and a matching phase that can be executed in real time. There are some of the factors involved in pattern matching performance, and strategies that you can use to achieve the best results.

You can use pattern matching in the following three general applications:

• Alignment - determines the position and orientation of a known object by locating fiducials. Use the fiducials as points of reference on the object.

• Gauging - measures lengths, diameters, angles, and other critical dimensions. If the measurements fall outside set tolerance levels, the component is rejected. Use pattern matching to locate the object you want to gauge.

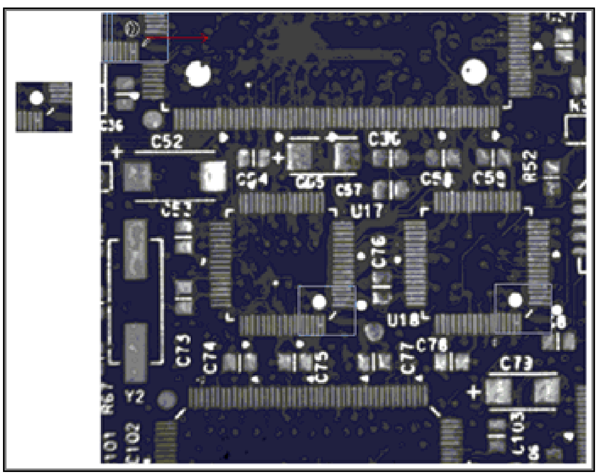
• Inspection - detects simple flaws, such as missing parts or unreadable print.

2. DESCRIPTION OF ALGORITHM

In this chapter is introduced the mechanism of pattern matching algorithm, the steps of algorithm realization, such as template learning and matching [1].

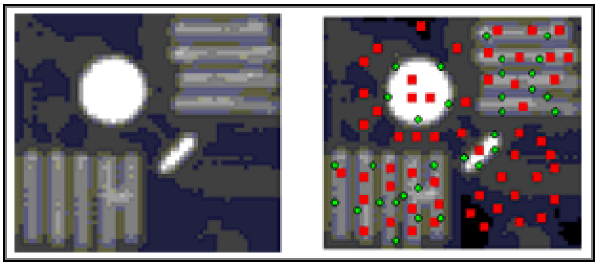
2.1. Learning Phase

The learning phase of pattern matching involves analyzing the template image to find features that can be exploited for efficient matching performance. This phase is what gives advanced pattern matching techniques dramatically improved performance compared to traditional grayscale correlation methods.



The traditional methods have no learning phase; the template is simply compared to every possible location in the image via a 2D correlation. This is very computationally intensive and involves many redundant calculations. Below is an example of a printed circuit board and template. After performing the correlation at every point in the image, two matches were found.

With a learning phase, you can incorporate sub-sampling of the template image to drastically reduce the number of calculations. Below is an example of a template that has been sub-sampled. The red squares capture information about the overall structure of the image, while the green circles identify features of the image that will be useful for localizing the match precisely.



Pattern matching algorithm incorporates the following features in the learning phase:

Pseudo-random sub-sampling: If you sub-sample the template by taking a uniform grid, it is possible to miss important features of the template, such as horizontal and vertical edges. Random sampling tends to produce clusters and open areas; having clusters of samples in the same area of the template is inefficient, while the unsampled areas may contain important information. The pseudo-random technique maximizes the uniformity of the sampling throughout the template without using a pre-defined grid.

Stability analysis: The pseudo-random sample pixels are analyzed by checking their surrounding neighborhood for stability (uniformity). Each pixel is classified according to how large its stable neighborhood is (that is, 3x3, 5x5, and so on). This information is helpful in reducing the number of comparisons needed in the matching phase.

Feature identification: Finally, an edge detection operation is performed on the template image. The resulting edge locations are retained for use in fine-tuning the location of the matched image.

Rotational-invariant analysis: If the user needs to be able to find a rotated version of the pattern in the search image, the learning phase also obtains information that can be used to detect rotation. This is done by identifying a circular intensity profile in the template image. A rotated version of the template has the same profile, but is shifted left or right by the amount of rotation.

This learning phase is quite complex, and the calculations can take up to several seconds to perform for large or complex templates. This phase only needs to be done once per template, however. The results can be stored in a file for use later in the actual pattern matching operation.

2.2. Matching Patterns

The matching phase uses the information from the learning phase to eliminate as much unnecessary calculation as possible. The matching algorithm used depends on whether the user has specified shift-invariant matching (finding the template at any location in the search image) or rotation-invariant matching (finding the template at any location AND rotation in the search image). Both are two-pass processes.

2.2.1. Shift-Invariant Matching

The first pass is a correlation that uses only the pseudo-randomly sampled pixels from the template image. The results of the stability analysis are used to determine how many positions in the search image can be skipped without missing any important features. For example, if all the sub-sampled pixels were found to be stable in a 3x3 neighborhood, the matching algorithm can skip two out of three correlations in each row and column while still guaranteeing that a match will be detected. This reduces the number of calculations required by a factor of 9. The first pass produces a number of candidate matches with rough position information.

The second pass only operates on the candidates identified in the first pass. The edge detection results of the learning phase are used to fine-tune the location of each match, and a score is produced for each based on the correlation result at that location. A user-provided score threshold determines which candidates are returned as matches.

2.2.2. Rotation-Invariant Matching

The first pass uses the circular intensity profile from the learning phase to search for shifted versions of that profile throughout the image. The user can input an allowable rotation range (in degrees) to reduce the number of calculations required in this pass. Several candidate matches are identified in this pass.

The second pass uses the pseudo-randomly sampled pixels to perform a correlation with all the candidates. A score is produced for each candidate to determine whether it should be classified as a match or not.

2.3. Techniques Of Pattern Matching

Pattern matching techniques include normalized cross-correlation, pyramidal matching, scale- and rotation-invariant matching, and image understanding.

2.3.1. Normalized Cross-Correlation

Normalized cross-correlation is the most common method for finding a template in an image. Because the underlying mechanism for correlation is based on a series of multiplication operations, the correlation process is time consuming. Technologies such as MMX allow for parallel multiplications and reduce overall computation time. To increase the speed of the matching process, reduce the size of the image and restrict the region of the image in which the matching occurs. Pyramidal matching and image understanding are two ways to increase the speed of the matching process.

2.3.2. Scale-Invariant and Rotation-Invariant Matching

Normalized cross-correlation is a good technique for finding patterns in an image when the patterns in the image are not scaled or rotated. Typically, cross-correlation can detect patterns of the same size up to a rotation of 5° to 10°. Extending correlation to detect patterns that are invariant to scale changes and rotation is difficult.

For scale-invariant matching, you must repeat the process of scaling or resizing the template and then perform the correlation operation. This adds a significant amount of computation to your matching process. Normalizing for rotation is even more difficult. If a clue regarding rotation can be extracted from the image, you can simply rotate the template and perform the correlation. However, if the nature of rotation is unknown, looking for the best match requires exhaustive rotations of the template.

You also can carry out correlation in the frequency domain using the Fast Fourier Transform (FFT). If the image and the template are the same size, this approach is more efficient than performing correlation in the spatial domain. In the frequency domain, correlation is obtained by multiplying the FFT of the image by the complex conjugate of the FFT of the template [3]. Normalized cross-correlation is considerably more difficult to implement in the frequency domain.

2.3.3. Pyramidal Matching

You can improve the computation time of pattern matching by reducing the size of the image and the template. In pyramidal matching, both the image and the template are sampled to smaller spatial resolutions. For instance, by sampling every other pixel, the image and the template can be reduced to one-fourth of their original sizes. Matching is performed first on the reduced images. Because the images are smaller, matching is faster. When matching is complete, only areas with high match scores need to be considered as matching areas in the original image.

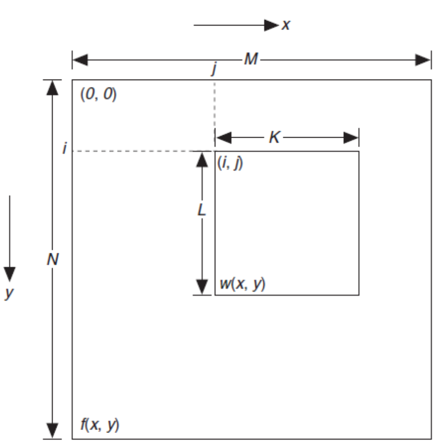
2.4. Normalized Cross-Correlation In-Depth

The following is the basic concept of correlation: Consider a sub-image w(x, y) of size K × L within an image f(x, y) of size M × N, where K ≤ M and L ≤ N. The correlation between w(x, y) and f(x, y) at a point (i, j) is given by

$$C(i,j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x,y) f(x+i, y+j)$$

, where i = 0,1,…M – 1, j = 0,1… N – 1, and the summation is taken over the region in the image where w and f overlap.

The figure below illustrates the correlation procedure.



Assume that the origin of the image f is at the top left corner. Correlation is the process of moving the template or sub-image w around the image area and computing the value C in that area. This involves multiplying each pixel in the template by the image pixel that it overlaps and then summing the results over all the pixels of the template. The maximum value of C indicates the position, where w best matches f. Correlation is not accurate at the borders of the image.

Basic correlation is very sensitive to amplitude changes in the image, such as intensity, and in the template. For example, if the intensity of the image f is doubled, so are the values of c. You can overcome sensitivity by computing the normalized correlation coefficient, which is defined as

$$R(i,j) = \frac{\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x,y) - \overline{w})(f(x+i,y+j) - \bar{f}(i,j))}{\left[\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x,y) - \overline{w})^2\right]^{\frac{1}{2}} \left[\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (f(x+i,y+j) - f(i,j))^2\right]^{\frac{1}{2}}}$$

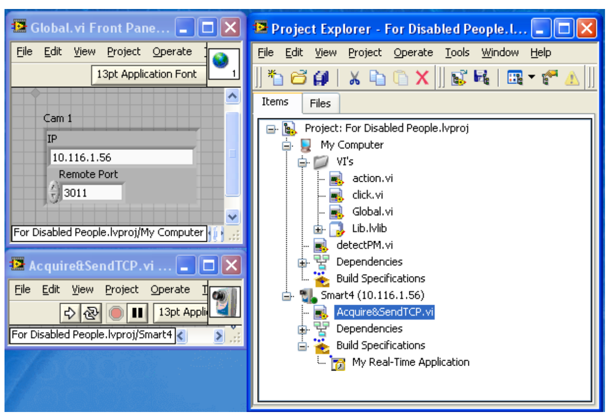
, where $\overline{w}$ (calculated only once) is the average intensity value of the pixels in the template w. The variable f is the average value of f in the region coincident with the current location of w. The value of R lies in the range –1 to 1 and is independent of scale changes in the intensity values of f and w.

3. REAL APPLICATION

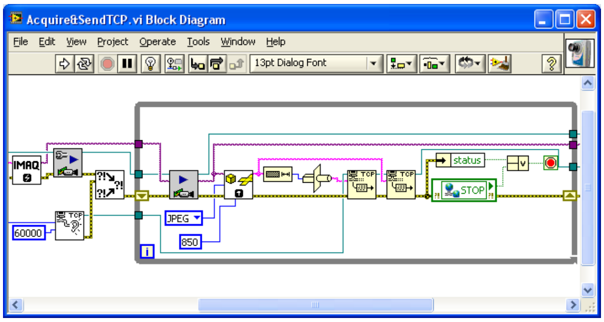
Real application works on images acquired from several types of cameras, such as GigE Cameras, USB/Web Cameras, Smart Cameras (Real-Time Targets). Here is introduced the application using Smart Cameras, which is Real-Time module and could be connected to the host using Ethernet or COM port.

3.1. Controlling PC Mouse Cursor With Smart-Camera

The project was done on National Instruments engineering product, called LabVIEW [4]. It's consists of two parts.

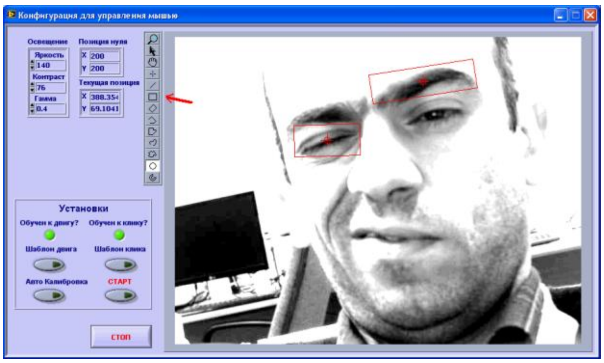


1. Real-Time part, which is sending acquired image via TCP/IP to the host computer. The program is compressing image data to JPG format before send to be able send more data in unit time. Below is the block diagram of that program.



2. Host part, which implement the analyzing of received image data via pattern matching algorithm and then send the results to mouse move controlling subVI's. It is possible to move mouse via face moving, left click and double click.

Front panel interface is on Russian, as it was done for Russia.



You can change some parameters of image being acquired, as it’s brightness, contrast and gamma, then you have to learn and set templates for mouse click and mouse move, you can do auto calibration of template position on an image, then start control mouse by moving your head.

In the image of front panel above you could see how the program works, it’s generate the click of mouse by closing eye and move the mouse depending on a position of human eyebrow.

ACKNOWLEDGEMENT

I would like to thank the company National Instruments Armenia with head of Aram Salatian for supporting me in my research, giving me required equipment and NI software products. Also I would like to say thank to the head of SPL department in IPIA Hakob Sarukhanyan

REFERENCES

[1] National Instruments Corporation, “NI Vision Concepts Manual”, 2000-2009.

[2] A. K. Jain Fundamentals of Digital Image Processing, Prentice-Hall Inc, Englewood Cliffs, 1989.

[3] H. Sarukhanyan, S. Agaian, Integer to Integer and Quantized Fast Fourier Transforms, *CSIT 2007*, pp. 204-207.

[4] National Instruments LabVIEW Development System, www.ni.com.